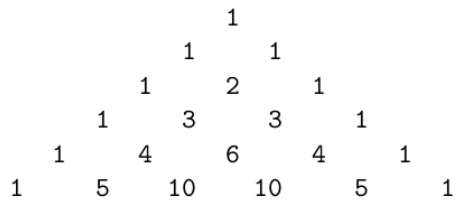

Programowanie IIR, 29.10.2021

Zadanie
Napisz kod wyznaczający wyrazy trójkąta Pascala do jego 13. wiersza. Do przechowania danych użyj 2-wymiarowych tablic statycznych. Wypisz wyrazy trójkąta na ekran stosując następujące formatowanie:

Zadanie 1
Napisz funkcję o prototypie:

```
double* aver (double* arr, size_t size, double& average);
```

która pobiera tablicę liczb typu double i jej wymiar oraz zmienną average typu double przez referencję. Zadaniem tej funkcji jest wstawienie do zmiennej average średniej arytmetycznej elementów tablicy oraz zwrócenie adresu tego elementu tablicy, którego wartość jest najbliższa tej średniej.
Przykład: następujący program

```
#include <iostream>
#include <cmath>

using namespace std;

double* aver (double* arr, int size, double& average) {
 // ...
}

int main ()
{
    double arr[] = {1, 7, 5, 4, 3, 2, 8, 9, -1, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    double average = 0;

    const double* p = aver (arr, size, average);
```

```
    cout << *p << " " << average << endl;

    return 0;
}
```

powinien wypisać:
4 4.3

Zadanie 3

Stwórz zestaw funkcji operujących na obiektach vector¡...¿:

- `bool allDiff (const vector<int>Tab)` – pobiera wektor danych i sprawdza, czy wszystkie jej elementy są różne.
- `int numDiff (const vector<int>& tab)` – pobiera referencję do wektora, a zwraca liczbę elementów unikalnych. Na przykład dla wartości [1, 5, 1, 1, 5] zwróci 2.
- `int fillWithPrimes (vector<int>&tab)` – pobiera wektor, wypełnia ją kolejnymi liczbami pierwszymi i zwraca ostatnią z nich.
- `size_t tabRem (vector<int>& tab, size_t from, size_t to)` – pobiera wektor i dwa indeksy, `from` i `to`; następnie „usuwa" elementy od tego z indeksem `from` (włącznie) do elementu z indeksem `to` (wyłącznie). Funkcja zwraca nowy wymiar wektora. Na przykład, jeśli fromjest 2, a to jest 4, to tablica [1, 2, 3, 4, 5, 6] będzie teraz zawierać na początku 1, 2, 5, 6, a zwrócone zostanie 4.

```
#include <iostream>
#include <vector>
using namespace std;

bool isPrime (int n) {....}
bool allDiff (const vector<int> tab) { ... }
int numDiff (const vector<int>& tab) { ... }
int fillWithPrimes (vector<int>& tab) { ... }
size_t tabRem (vector<int>& vec , size_t from, size_t to) { ... }

int main()
{
    vector<int> Tab = {3, 2, 3, 2, 5};
    cout << "allDiff: " << boolalpha << allDiff(Tab) << endl;
    cout << "numDiff: " << numDiff (Tab) << endl;
    vector<int> TabPrimes (15, 0);
    auto LastPrime = fillWithPrimes (TabPrimes);
    cout << "Primes: ";
```

```
    for (auto p : TabPrimes)
     cout << p << " ";
    cout << endl;
    cout <<"Last: " << LastPrime << endl;
    vector<int> TabBlock = {1, 2, 3, 4, 5, 6, 7};
    cout << "Original vec: ";
    for (auto x : TabBlock)
     cout << x << " ";
    cout << endl;
    auto newDimh = tabRem (TabBlock , 2, 5);
    cout << "After removing: ";
    for (auto x : TabBlock)
     cout << x << " ";
    cout << endl;

    return 0;
}
```

powinien wydrukować:
allDiff: false
numDiff: 3
Primes: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
Last: 47 Original vec: 1 2 3 4 5 6 7
After removing: 1 2 6 7

Dodatkowe:
Zadanie 4
Napisz `eratostenes` znajdujący wszystkie liczby pierwsze mniejsze od danej liczby naturalnej `nmax` metodą sita Eratostenesa.

Program powinien czytać tę liczbę ze standardowego wejścia.

Zadanie 5
Napisz funkcję sortującą wektor metodą bombelkową. Funkcja ma zmieniać wektor.